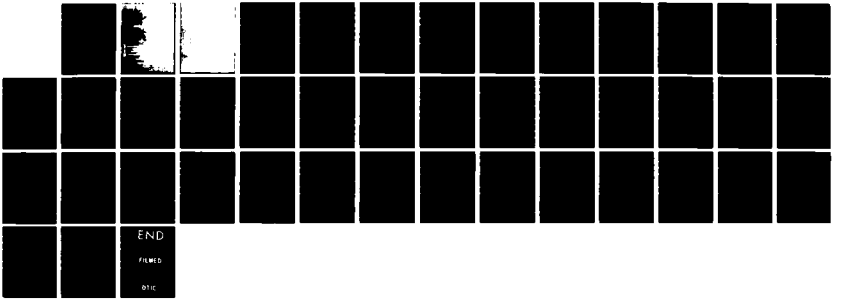
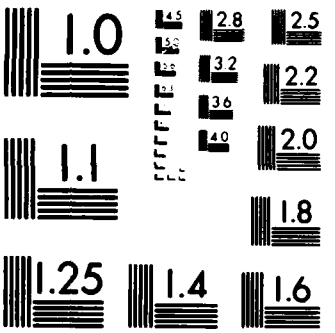
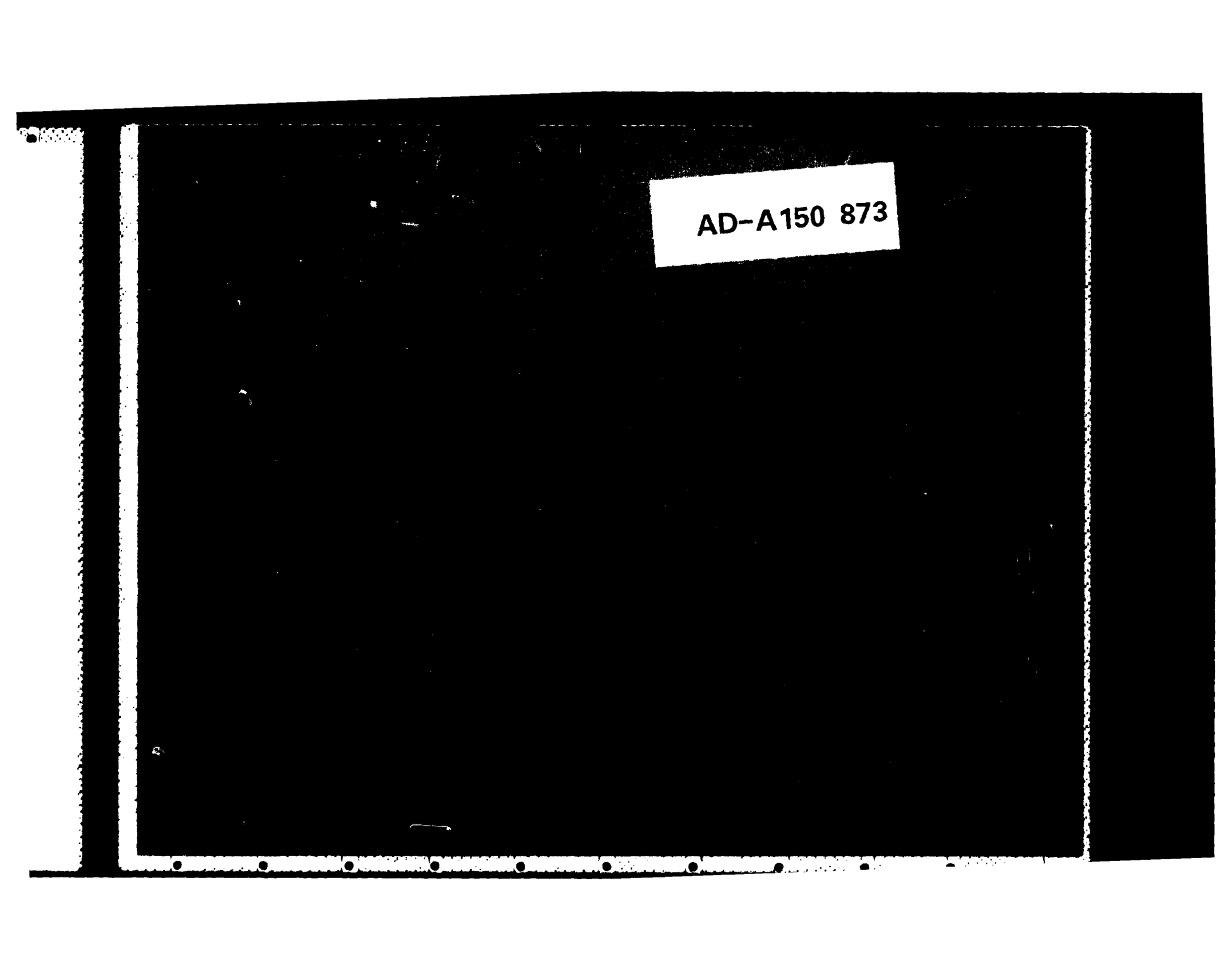
AD-A150 873 COMPARATIVE ARCHITECTURES FOR A MULTIPLE FUNCTION SPEECH PROCESSOR(U) MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB E SINGER 18 DEC 84 TR-712 1/1

UNCLASSIFIED ESD-TR-84-283 F19628-85-C-0002 F/G 17/2 NL







MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

# COMPARATIVE ARCHITECTURES FOR A MULTIPLE FUNCTION SPEECH PROCESSOR

*E. SINGER*

*Group 24*

TECHNICAL REPORT 712

18 DECEMBER 1984



LEXINGTON MASSACHUSETTS

## ABSTRACT

A study of the feasibility of realizing a compact, multiple function speech processor using digital signal processing integrated circuits is presented in this report. The processor is required to accommodate a 2400 bps Linear Predictive (LPC-10) vocoder, a 9600 bps Adaptive Predictive (APC) vocoder, and wireline modems at the corresponding data rates. Architectures employing the Texas Instruments TMS32010 and the Fujitsu MB8764 digital signal processing integrated circuits appeared to be most likely to achieve this goal. Since the current generation of DSP chips does not permit the algorithms to be contained in a processor simultaneously, the study was confined to the realization of each of the functions individually using replaceable ROMs. A proposed architecture based on the TMS32010 includes automatic address generation hardware to permit fast processing of large blocks of data stored in external memory. Although specifically tailored to implement the Adaptive Predictive Coding algorithm, the processor appears to be a likely candidate for the multiple function task as well. An architecture based on the MB8764 includes special purpose hardware which adds an interrupt capability to the device. A detailed study of the Fujitsu-based design indicates that a compact realization of the multiple function processor can be realized using commercially available hardware and a custom IC fabricated using in-house design tools.

CONTENTS


| | |
|---|---|
| ABSTRACT | iii |
| LIST OF ILLUSTRATIONS | vi |
| LIST OF TABLES | vi |
| I. INTRODUCTION | 1 |
| II. ALGORITHMS | 2 |
| III. DEVICES | 8 |
| IV. ARCHITECTURE USING THE TMS32010: THE APC PROCESSOR | 12 |
| V. ARCHITECTURE USING THE MB8764 | 19 |
| VI. SUMMARY | 30 |
| ACKNOWLEDGEMENT | 34 |
| REFERENCES | 35 |

## LIST OF ILLUSTRATIONS


| | | |
|---|---|---|
| 1. | 9600 bps modem block diagram. Buffer sizes are shown in parentheses. | 5 |
| 2. | APC processor architecture. | 15 |
| 3. | MB8764-based speech processor architecture. | 23 |
| 4. | Peripheral interface operation. | 25 |
| 5. | Functional realization of interrupt controller. | 26 |
| 6. | Component level implementation of MB8764-based speech processor. | 28 |
| 7. | Proposed processor layout. | 31 |


## LIST OF TABLES


| | | |
|---|---|---|
| I | *Performance Comparison of Two DSP Chips* | 9 |
| II | RAM Allocation for APC Processor | 16 |
| III | Critical Loop Execution Times for APC Processor | 17 |
| IV | Modem Inner Loop Computations | 20 |
| V | Speech Processor Components | 32 |

# I. INTRODUCTION

The increasing sophistication of VLSI digital signal processing integrated circuits has provided a means for the realization of complex speech processing algorithms in compact, low power equipments. The utilization of these devices for the implementation of a multiple rate speech terminal is the subject of this report. The algorithms to be incorporated in such a terminal include a 2400 bps DoD-interoperable Linear Predictive Coder, a 9600 bps DoD-interoperable Adaptive Predictive Coder, and wireline modems at the commensurate data rates. Since a compact implementation of a full multi-rate speech terminal may be beyond the capabilities of contemporary digital signal processing ICs, this study focused on the feasibility of designing a flexible speech processing board whose function can be customized to a single task by inserting the appropriate set of PROMs.

Several speech processor designs based on DSP chips have been reported recently. A compact processor developed at Lincoln Laboratory relies on three Nippon Electric Company μPD7720 signal processing interface microcomputers to implement a full duplex LPC vocoder [1]. Other compact, low power realizations of LPC-10 vocoders have been reported by groups at Texas Instruments [2] and the Canadian Department of Communications (CDC) [3] using the Texas Instruments TMS32010 microcomputer. The CDC implementation uses 20 chips, dissipater 2W of power, and requires 96% of real time for full duplex operation. The TI implementation uses a different architectural approach and runs in 87% of real time using about 20 chips. A compact architecture designed to implement a 9600 bps APC

algorithm using the TMS32010 has also been proposed [4]. The study estimated that about 85-90% of real time would be required for full duplex operation. Although these real time margins are fairly narrow, the results suggest that a TMS32010-based design would be a potential candidate for the multi-function processor task. More recently another single-chip microcomputer, the Fujitsu MB8764, has become available [6]. Because of its significantly improved computational capabilities, an architecture using this device was also studied as a means of realizing the multiple function speech processor.

The present state of DSP IC technology does not yet permit the design of a compact, general purpose speech processor capable of realizing arbitrarily complex speech algorithms in real time. The processor must generally be tailored to the speech algorithms of interest and must seek to optimize its performance within the limits imposed by a chip's computational capability, memory access, and I/O structure. This report describes a study in which architectural descriptions of speech processors using both the TMS32010 and the MB8764 were developed and compared for the task of implementing the desired speech processing functions. Detailed descriptions of the relative capabilities of these devices will be presented with emphasis on their suitability for performing the speech processing functions of interest.

## II. ALGORITHMS

Hardware realizations of real time Linear Predictive Coding and Adaptive Predictive Coding algorithms have been the subject of many studies and the issues regarding their implementation are well understood [1-5].

The major computational elements in an autocorrelation-type LPC vocoder are the calculation of the autocorrelation coefficients, the pitch detection, and the synthesizer filter. An APC vocoder, which is computationally more demanding, requires direct form filters to implement the pitch and spectral prediction filters at the analyzer and synthesizer as well as a pitch detector and low order autocorrelation analysis. The approximate execution times of nearly all of these signal processing functions can often be readily determined by generating the actual software required by the processor. Benchmark execution times for FIR filters and FFTs are invariably provided by the DSP chip manufacturers. Unfortunately, these figures usually assume certain idealized conditions which may not be applicable to speech processing without the addition of generous amounts of external hardware. These issues will be explored more fully in subsequent sections. Furthermore, routines such as the Gold pitch detector are decision oriented and cannot be realized with a tight loop. Evaluation of the DSP chip performance for such routines requires the optimization of large blocks of code and may be a time consuming task.

Another important aspect of a signal processing algorithm which can affect the speech processor architecture is the amount of memory required to implement the algorithm. Data memory requirements will often vary with the methods used to process the data. For example, the objective of both the LPC and APC analyzers is the representation of a frame of speech data at a reduced data rate. In LPC analysis, double buffering of the input data may be avoided by updating the autocorrelation coefficients as part of the foreground processing. Since the coefficients are usually computed as

double length words on a 16-bit machine, two memory reads, a multiply accumulate, and two writes are required to update each coefficient at each sampling instant. In Adaptive Predictive Coding, buffering of the input data is unavoidable since the parameters used to characterize the frame of speech are then applied to the speech samples themselves. In fact, the presence of a pitch prediction filter in the analysis loop, with its relatively long memory, usually necessitates the presence of a triple buffer of input data. Randolph has determined that 2K of data RAM is required to implement a 9600 bps APC algorithm [4], whereas real time LPC has been implemented using only 512 words of RAM [5].

A block diagram of the 9600 bps wireline modem algorithm evaluated in this study is shown in Fig. 1 [7]. Serial data from the vocoder are made available to the modem over a digital data link. A differentially encoded quadrature amplitude modulation (QAM) scheme is used to modulate a carrier by a shaped pulse with one of 16 possible amplitude-phase combinations. The amplitude-phase combination is specified uniquely by a 4-bit cluster in the input data stream. Since messages are signalled 2400 times per second the overall data rate capability of the signalling scheme is 9600 bps. The encoded signal is converted to analog form and low-pass filtered to 3600 Hz. The received signal is oversampled to permit baud synchronization to be maintained without adjusting the sampling clock. The signal is then scaled by the automatic gain control routine in order to provide the equalizer with a signal of uniform level. The adaptive equalizer is designed to compensate for magnitude and phase distortions introduced by the channel which would cause intersymbol interference in the received waveform and reduce the fidelity of the decoding process. The equalizer

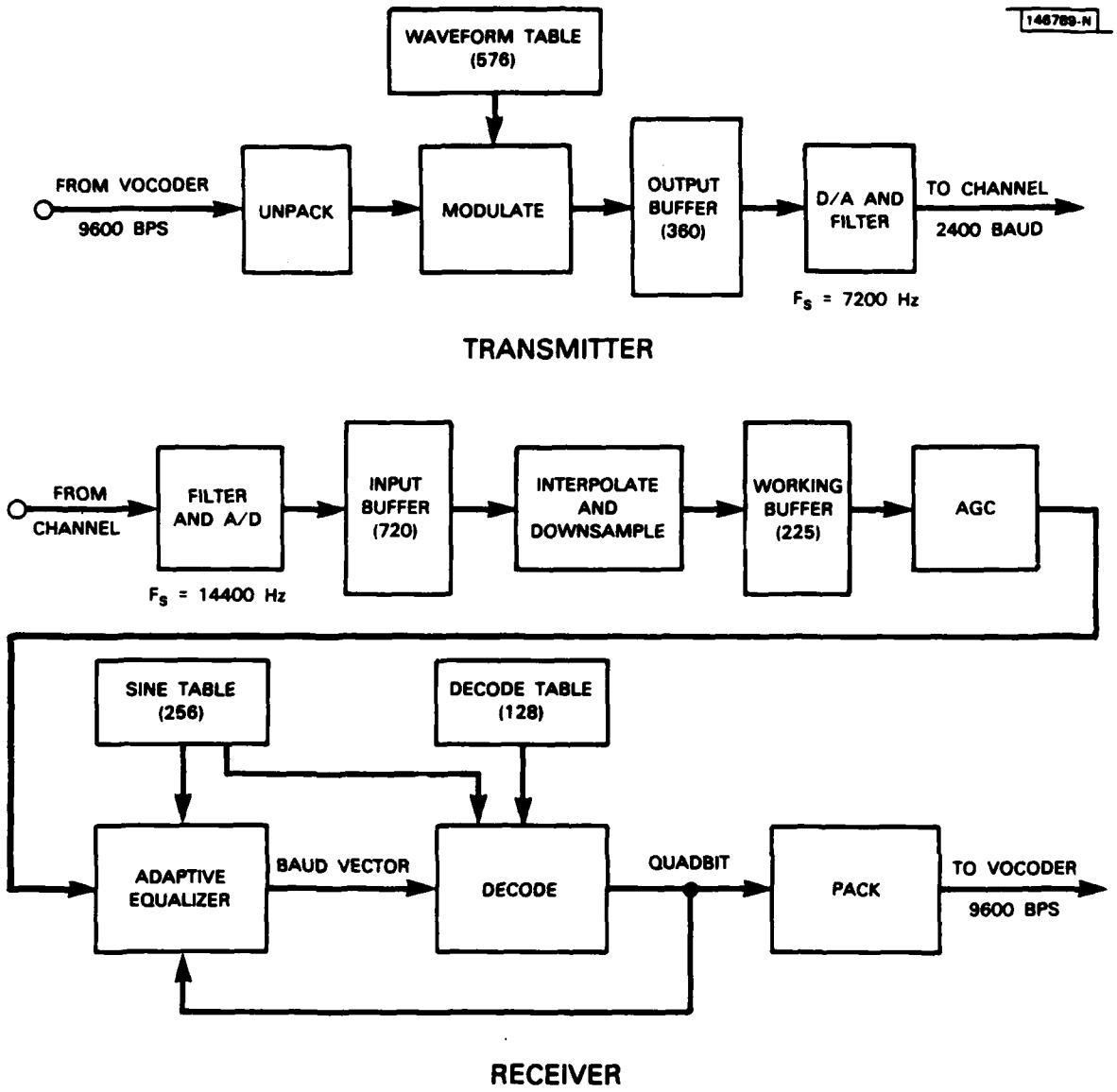



Fig. 1. 9600 bps modem block diagram. Buffer sizes are shown in parentheses.

consists of an in-phase and a quadrature filter whose coefficients are updated using feedback from the decoder. The decoder uses the vector output of the equalizer and decides which of the 16 possible messages was received. Decoding occurs at a 2400 Hz rate and produces a 9600 bps digital signal which is made available to the vocoder synthesizer.

The function of the decoder is to convert each complex output sample of the equalizer into one of the 16 allowable 4-bit combinations. The rectangular coordinates produced by the equalizer are converted to polar form and a locally generated carrier phase reference is subtracted from the phase component. Jitter in the received signal is estimated and removed by a jitter predictor. The received magnitude and phase are quantized and the error in the resulting phase is used to update the adaptive equalizer coefficients.

The modem functions described above contain the bulk of the computations performed during normal operation. Other routines which are essential to the full description of the modem but do not have a major impact on either computation or storage include training, initialization, baud sync acquisition, scrambling and unscrambling.

A careful examination of the modem storage requirements indicates that roughly 1.5K words of RAM are necessary for data buffering and 1K words of ROM for the tables. The bulk of the data RAM is required for double buffering the received data after A/D conversion. As usual, the data storage requirements can be reduced by restructuring the interface between the modem and the vocoder. Similarly, the ROM requirements can be reduced by storing only a portion of the modulation table and by computing the

remainder of the values on the fly, thereby trading off memory requirements for execution time. For the purposes of this study it is assumed that 2K of RAM and 1K of data ROM suffice to implement the 9600 bps modem.

The most computationally demanding element of the modem is the adaptive equalizer which consists of two 48-tap FIR filters. Since the sampling rate is 7200 Hz at the filter input and 2400 Hz at the output, the total number of operations required is 230,400 multiply-accumulates per second. A real time implementation of the 9600 bps modem on a high speed, general purpose signal processor indicated that approximately 40% of the total execution time is devoted to the filtering [8]. This figure will be used as a guideline to judge the overall processing capabilities of the architectures to be presented later.

Contemporary DSPs are generally designed to implement sums-of-products routines in an extremely efficient manner and can often outperform general purpose processors running at much higher speeds. These computational capabilities are particularly advantageous in speech processing applications requiring extensive FIR filtering and correlation calculations. Effective exploitation of this capability requires rapid and ready access to large blocks of data without the necessity of storing intermediate results. Unfortunately, the DSP chips currently available provide limited on-board RAM and often exact unacceptable penalties in execution time for off-chip access of data. Thus, the utilization of DSPs in sophisticated speech processing applications may require not only a careful study of the processing routines but also the design of external hardware which will allow the processor to operate more efficiently.

This factor will greatly influence the architectural design of the speech processor and will be discussed more fully in subsequent sections.

## III. DEVICES

Because the computational demands of speech processing algorithms are substantial, only the most sophisticated of the available digital signal processing chips were considered in this study. Randolph [4] determined that the Texas Instruments TMS32010 was most suitable as the signal processing engine of a single-chip APC speech processor when compared to either the Nippon Electric Company μPD7720 or the AT&T Bell Labs DSP I. The latter two devices were rejected primarily because of their limited I/O capability and lack of convenient access to sufficient amounts of data RAM. More recently the Fujitsu MB8764 has become available and appears to offer significant computational advantages over its competitors.

Table I presents a comparison of some of the important features of the TMS32010 and the MB8764. The internal architectures provide high speed, pipelined multiply-accumulators which permit computation of high accuracy correlations and filters. Effective use of these resources requires that the data be available in block form. Consequently, foreground computation of an autocorrelation or filter function on a sample-by-sample basis makes inefficient use of the device. Furthermore, of the 26 bits of accumulator provided by the Fujitsu chip, only the upper 16 can be read directly. Storing an extended precision word in a time-critical routine is obviously impractical. As a result, double buffering of data is generally advisable for any loop requiring repeated use of the multiply-accumulator.

TABLE I

PERFORMANCE COMPARISON OF TWO DSP CHIPS

| Feature | MB8764 | TMS32010 |
|---|---|---|
| POWER | 300 mW | 900 mW |
| INSTR. CYCLE | 100 ns (no off-chip RAM), 125 ns (45 ns RAM) | 200 ns |
| MULTPLY/ACC<br>-Precision<br>-Speed (ideal) | <br>26 bits<br>1 cycle | <br>32 bits<br>2 cycles |
| RAM (internal) | 256 words | 144 words |
| ADDR. SPACE (external) | 1K data RAM, 1K ROM | 4K total |
| ADDRESSING MODES | direct, indexed | direct, indirect |
| INSTRUCTION SET | compound (24 bit word) | simple (16 bit word) |
| I/O | no interrupt | interrupt |

A minimum of two cycles (400 ns) is required by the TMS32010 for each multiply-accumulate. A loop containing a single multiply-accumulate operation and conditional branch takes four cycles since the branch instruction itself requires two. A common approach when programming the TMS32010 is to write the operations in straight-line code rather than in a loop in order to eliminate the branching overhead. Obviously, this method of achieving maximum computational throughput requires more program memory.

An important advantage of the Fujitsu MB8764 is that it is possible to perform a multiply-accumulate operation in a single cycle if the two operands are stored in separate 128-word sections of internal RAM (referred to as ARAM and BRAM). This is made possible by providing separate paths from the memories to the multiplier inputs and by having an extra stage of pipelining in the multiplier. Furthermore, with a cycle time of only 100 ns, the Fujitsu DSP can potentially provide increased computational throughput compared to the TMS32010. As with the TMS32010, jump instructions conditioned on a loop counter take two cycles to execute.

As Table I indicates, a serious shortcoming of present generation signal processing chips is the inadequate amount of on-chip RAM for performing speech processing functions (144 words in the TMS32010 and 256 words in the MB8764). Any portion of the 4K external program memory addressable by the TMS32010 can also be employed as data RAM using the 3-cycle instructions TBLR (table read) and TBLW (table write). For example, a TBLR instruction will transfer a word from external RAM at a location specified by the contents of the accumulator to a location in internal RAM given in the instruction field. In reality, more than 3

cycles are often necessary for external memory access in order to accommodate saving and restoring the accumulator contents. Faster off-chip memory access can be achieved by treating a memory transfer as an I/O operation which requires only 2 cycles on the TMS32010. For this configuration, external hardware must be included to provide the proper address to the RAM automatically. In fact, most proposed vocoder architectures using the TMS32010 incorporate some form of automatic address generation.

Although on-chip RAM in the MB8764 is also limited, hardware has been incorporated in the chip to permit addressing 1K of external RAM within a single machine cycle. Since external RAM is treated as an extension of BRAM, the one cycle multiply-accumulate can still be achieved as long as it is feasible to store one of the operand buffers internally in ARAM. However, operating with external RAM imposes an execution time penalty on the processor since settling and set-up times of the chip must now be met. Use of external RAM with an access time of 45 ns would result in an instruction cycle of about 125 ns. Both the 1K of external data memory and the 1K of program memory provided by the MB8764 are inadequate for a vocoder implementation.

The MB8764 contains several other features which may enhance the convenience of the device's use. The chip provides two index registers for use in indexed addressing whereas the TMS32010 offers none. The Fujitsu chip has two loop counters, one 8 bits wide and one 4 bits wide, to facilitate instruction loops. The chip also provides a one level stack for each of the index registers to improve nested loop run times. However,

since the stack can only be controlled by special purpose instructions, the usefulness of this feature is limited.

Perhaps the most serious shortcoming of the MB8764 is that it contains no hardware interrupt. Thus, the software must be structured so as to poll peripheral devices at judiciously chosen intervals to determine whether any device requires service. Two flags are provided on the chip for this purpose and can be set externally and tested by the software. However, polling places an additional burden on the programmer and adds to the execution time. A rough calculation shows that the overhead required by the software polling is at least 5% of real time for a 2400 bps vocoder.

It is clear that architectures using either of these chips must attempt to overcome their shortcomings for speech applications. A speech processor designed around the TMS32010 needs to include support hardware which permits rapid (2 cycle) access to external memory. A Fujitsu-based processor must employ an architecture which eliminates the overhead associated with software polling and permits the chip to operate at its maximum computational capability. Speech processor architectures designed using these guidelines will be presented in the following sections.

## IV. ARCHITECTURE USING THE TMS32010: THE APC PROCESSOR

A candidate architecture for a multi-function speech processor is the design developed at Lincoln Laboratory for a TMS32010-based Adaptive Predictive Coder [4]. Although designed specifically for the APC task, the processor incorporates the features necessary to achieve high execution speeds using the TMS32010 and was therefore studied for its applicability to the multiple function processor discussed in this report. The

suitability of the APC processor to more general usage depends on the extent to which the special features of the hardware can be beneficially employed for a given speech processing function.

A functional block diagram of the architecture is shown in Fig. 2. The major feature of its architecture is the addition of special purpose hardware which permits rapid transfer of data blocks between the device and external RAM. As discussed earlier, the TMS32010 provides special instructions which permit access of external memory at an address given by the low 12 bits of the accumulator. Although the instructions require 3 cycles to execute, the effective execution time is longer since the address to external memory must be loaded into the accumulator and any other data previously residing in the accumulator must be saved and restored. Such a process generally takes at least 7 cycles. If a double precision accumulation is being performed, two words must be saved and restored at every sample time. This procedure is inefficient for fast processing of blocks of data and cannot take advantage of the fast multiply-accumulate feature.

A more efficient method of addressing external memory involves the use of the TMS32010 I/O ports. Data can be transferred through one of the eight ports provided by the device via software instructions which take two cycles each to execute. Data transfers to external RAM can thus be reduced to two cycles if some means of automatically generating the necessary addresses is provided. Since the intensive computations performed in speech processing algorithms usually involve sequential access to blocks of data, additional logic can be included in the processor to set initial addresses

and perform autoincrementing. The external memory controller and address multiplexer shown in Fig. 2 incorporate the hardware necessary to implement these operations. Automatic addressing is accomplished through the use of 2 Advanced Micro Devices Am2940 programmable address generators. The devices are initialized at the beginning of a routine via a dedicated data port and increment their addresses after two data words (the operands in a multiply-accumulate, for example) have been fetched. Incrementing occurs without further program intervention.

The processor in Fig. 2 is carefully tailored to perform the specific functions required by an Adaptive Predictive Coder. In particular, the 2K words of RAM have been partitioned specifically to accommodate APC processing functions. The RAM allocation scheme is described in Table II. The first 512 words of RAM can be accessed sequentially via two I/O ports to allow computation of the pitch filter coefficient and the predictive quantization loop. A special increment-by-four option is provided for this portion of RAM to allow fast AMDF pitch period calculation. The next two 256-word blocks of RAM are implemented individually as circular buffers. These blocks are used to store the reconstructed speech for the analyzer and synthesizer. The remaining 1K of RAM and 2K of ROM can only be accessed using the table read/write instructions.

Table III presents the execution times required by the APC processor to perform critical processing routines [4]. Although these estimates indicate that a 9600 bps APC could be implemented in about 86-90% real time, they do not include routines for coding and decoding, block data transfers, synchronization, scrambling, and error correction. Although

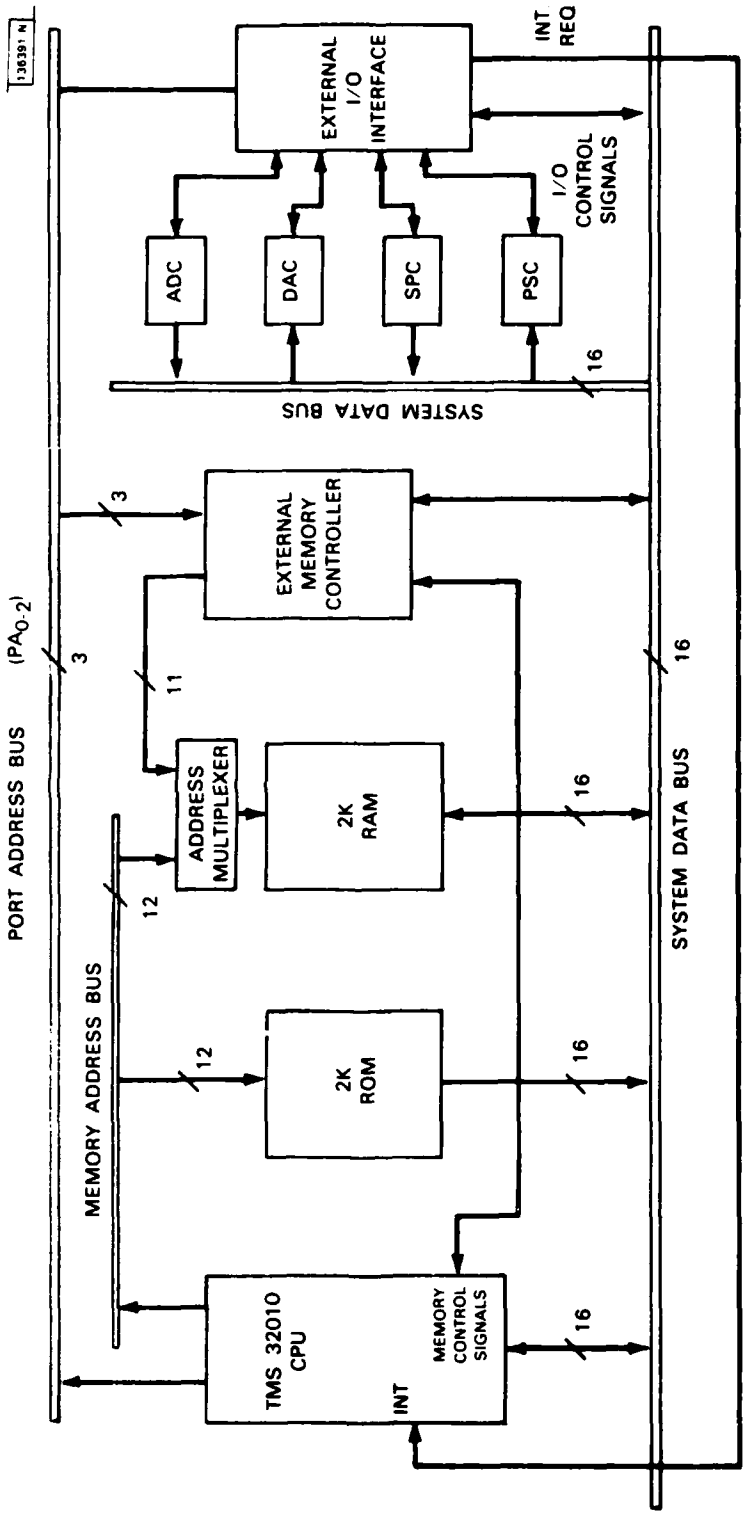



Fig. 2. APC processor architecture.

TABLE II

RAM ALLOCATION FOR APC PROCESSOR

| Page (256 words) | Buffer | Access |
|---|---|---|
| 0,1 | contiguous, auto-increment by 1 or 4 | IN/OUT |
| 2 | circular, auto-increment | IN/OUT |
| 3 | circular, auto-increment | IN/OUT |
| 4-7 | general | TBLR/W |

TABLE III

CRITICAL LOOP EXECUTION TIMES FOR APC PROCESSOR

| Operation | Execution Time (msec) | % Real Time |
|---|---|---|
| AMDF pitch estimation | 6.6 | 33.0 |
| Pitch coefficient | 0.62 | 3.1 |
| First residual and LPC autocorrelation | 2.92 | 14.6 |
| Reflection coefficients | 0.16 | 0.8 |
| Predictive quantization | 1.4-2.16 | 7.0-10.8 |
| Receiver | 1.6 | 8.0 |
| AD-DA I/O | 1.09-1.28 | 5.4-6.4 |
| Serial I/O | 1.4 | 7.0 |
| TOTAL | 17.19-18.14 | 86.0-90.7 |
| Autocorrelation pitch | 4.1 | 20.52 |
| TOTAL | 14.69-15.64 | 73.31-78.52 |

these routines are performed at the frame rate, the additional computational burden decreases the real time margin to an uncomfortable level. Additional margin can be obtained by replacing the AMDF pitch detector with an autocorrelation pitch detector. By using the special multiply-accumulate instructions available on the TMS32010 the amount of real time required to estimate the pitch period can be reduced to about 21% if only every fourth sample is used in the computation. Whether or not this method is valid in the context of the autocorrelation technique remains to be confirmed.

Successful use of the APC processor architecture for general purpose speech processing depends on whether or not a given algorithm can be structured to make effective use of the processor's special purpose features. An LPC-10 autocorrelation vocoder can be implemented efficiently in this processor by using the first 512 of RAM words as a double buffer for the input speech samples. Generation of the autocorrelation coefficients can then be performed via sequential I/O through the TMS32010 data ports and the fast multiply-accumulate instructions. Whether the 2K of ROM allocated for both program and data memory is adequate is an open question. A recent implementation of a TMS32010-based LPC vocoder which made heavy use of in-line coding to maximize throughput required 3K of ROM [3]. At this stage it is simply not possible to project accurately the amount of program ROM required to perform the speech processing functions of interest.

To determine the suitability of the APC processor for implementing the 9600 bps modem, the computational requirements of the modem were evaluated

and are presented in Table IV. The routines were selected because they involve highly repetitive computation and thus lend themselves to relatively straightforward evaluation. Other routines which are principally decision oriented were not closely studied primarily because of the software development effort involved. The table indicates that a substantial amount of computation is devoted to the adaptive equalizer which consists of two 48-tap FIR filters. The last section of the table presents an estimate of the amount of real time required for the TMS32010-based APC processor to execute the routines listed. A range of values is given to indicate that execution time is dependent on such factors as data location (on-chip vs. off-chip) and software structure (loop vs. in-line). Computation for the adaptive equalizer requires approximately 30% of real time if both the data and the filter coefficients are stored in external memory and if branching overhead is reduced by including 4 multiply-accumulates in a loop. With in-line code, the real time requirement drops to 27% but about 400 words of program memory are needed. If 96 of the 144 words of internal RAM are used for storing the filter coefficients, the real time requirements are reduced to 18%-21%. As mentioned earlier, adaptive equalization accounts for approximately 40% of the modem execution time on a general purpose processor.

## V. ARCHITECTURE USING THE MB8764

The use of the Fujitsu digital signal processing chip for speech processing applications appears particularly attractive. Its pipelined multiplier, parallel on-chip memories, and fast cycle time allow block correlations and filter computations to be performed very efficiently. In

TABLE IV

MODEM INNER LOOP COMPUTATIONS

| Routine: Name and Function | Adds/sec | Mult/sec | Div/sec | Real Time (MB8764) | Real Time (TMS32010) |
|---|---|---|---|---|---|
| MODULT -modulate | 21600 | | | 1.08% | |
| INTERP -interpolate | 7200 | 14400 | | 0.5% | 2.7-4.5% |
| AEQUAL -adaptive equalizer | 230400 (mult-acc) | | | 2.9-7.2% | 18.4-30.0% |
| ARCTAN -rectangular-to-polar | 4800 | 14400 | 2400 | 1.0% | 2.78% |
| JITTER -jitter prediction | 19200 | 12000 | | 0.69% | |
| ADAPT -update coefficients | 58800 | 58800 | | 3.07% | 7.73-10.0% |
| BSYNC -baud sync | 59400 | 61950 | 750 | 1.4-2.04% | 7.83-14.7% |

addition, provisions have been made to permit up to 1K of external RAM to be accessed in a single cycle. The second to last column of Table IV lists the execution times required by an MB8764-based processor to perform the modem routines assuming a 125 ns cycle time. With all the data stored off-chip, 7% of real time is used to perform the adaptive equalization. If the 96 FIR filter coefficients are stored internally in ARAM, only 4% of real time is required. The use of in-line code reduces this figure to 2.9%. The device's applicability for real time signal processing is hampered, however, by its lack of an on-chip interrupt handling capability and by the provision for no more than 1K words apiece of off-chip program and data memory. In this section, an architecture for an MB8764-based processor is proposed which employs special purpose hardware designed to permit interrupt driven communication between the DSP and its peripherals and to allow external program and data memory to be expanded to 4K words each.

The sequence of operations which occurs during an interrupt consists of halting normal program execution, saving the address of the next instruction to be executed, jumping to a predetermined location to service the interrupting device, and resuming the normal execution sequence at the saved program memory location. Obviously, it is critical that the state of the DSP be restored to the identical state it was in before interruption. In the MB8764, the 4-bit loop counter (CI) and the index register stack (XS and YS) either cannot be read or written to directly. Thus, sequences making use of these registers must not be interrupted. More importantly, successive instructions which load the multiplier cannot be interrupted due to the internal multiplier pipelining. Fortunately, tight loops consisting

of sequential multiplies contain instructions that can be interrupted. All of the software written for the routines of Table IV can accommodate interrupts without overhead.

The architecture proposed for an MB8764-based speech processor is shown in Fig. 3. Special purpose hardware has been added to permit the device to service I/O requests via interrupts. Requests for service are stored and prioritized by the I/O Request Handler which produces an enable signal during instruction sequences where interrupts are permitted. An I/O jump signal is then received by the Interrupt Controller which forces a jump instruction into the DSP and causes the internal program counter to be set to a preassigned vectored location. At the same time, the Interrupt Controller saves the current program address so that normal background processing can be resumed upon completion of the interrupt service routine. The I/O return signal forces another jump instruction into the DSP with the address field containing the stored return address. The return address stored by the MB8764 on its program counter stack cannot be used in the proposed interrupt scheme because its value is one greater than is needed at the end of an interrupt service routine.

An important advantage of the MB8764 is that it permits both program and data memory to be stored off-chip and accessed in a single cycle. Since the allowable 1K of memory is insufficient for the speech processing functions of interest, provisions have been made to accommodate 4K each of program and data memory by using a paging scheme. The data page bits, program page bits, interrupt enable signal, and I/O return signal are part of an auxiliary control field which is appended to each microinstruction

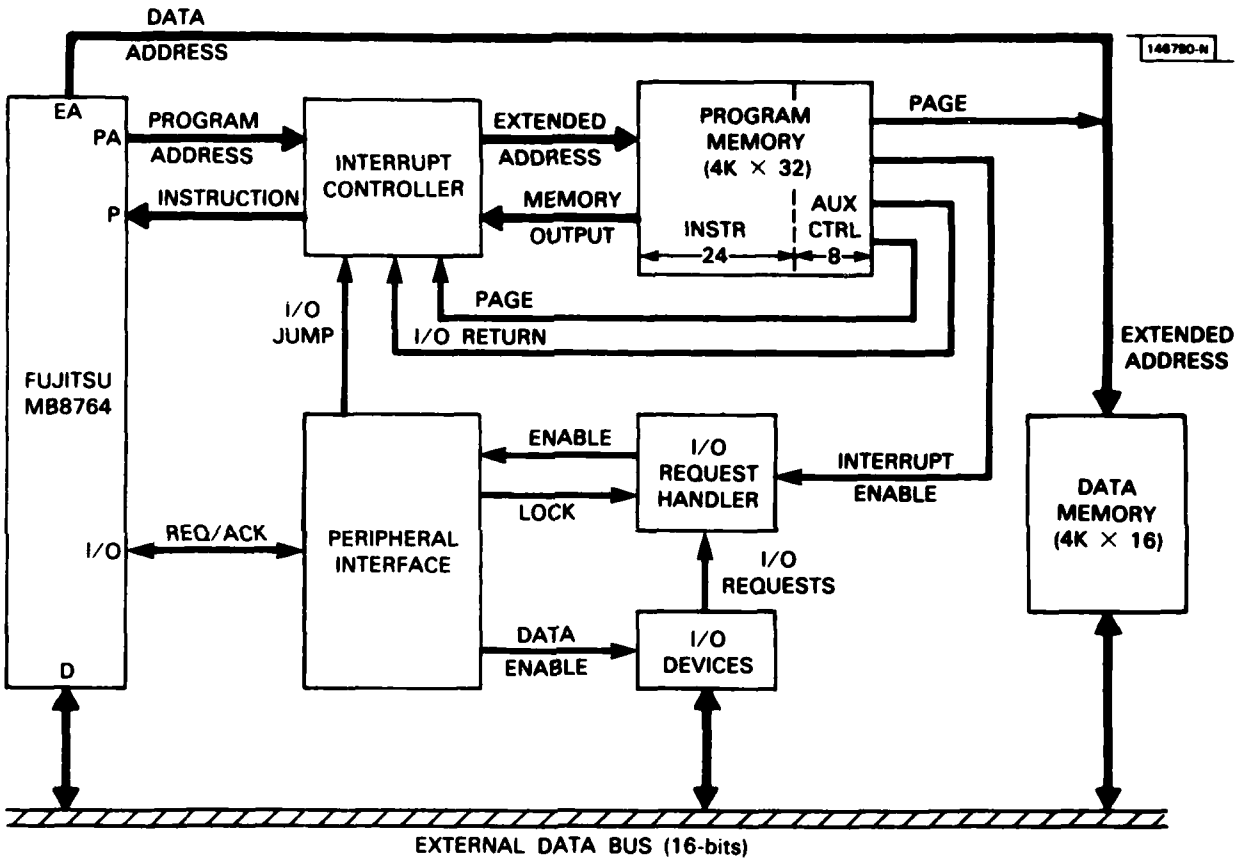



Fig. 3. MB8764-based speech processor architecture.

word and stored in a 4K x 8 PROM. Because of the availability of very dense high speed ROM, this method of providing control signals is an attractive alternative to the use of special paging and control registers.

The last major component of the MB8764-based processor architecture is the Peripheral Interface circuit. Its function is to provide the necessary handshaking signals to the DSP chip and the data enable signals to the I/O devices. A flow diagram of the sequence of events which occurs during a data transfer to and from a peripheral is shown in Fig. 4. In addition, the Peripheral Interface provides the jump signals used to initiate an I/O jump instruction and to save the value of the current address and page.

The functions of the Interrupt Controller are shown in detail in Fig. 5. During normal operation the 2 program page bits and the 24 instruction bits are derived from the output of program memory. When an interrupt occurs, the I/O jump control signal $\overline{\text{IOJ}}$ causes the Instruction Select hardware to place a hardwired jump instruction onto the P input of the DSP. This forces the DSP's internal program counter to a vectored location specified by the two vector address output bits (V) bits of the I/O Request Handler. When $\overline{\text{IOJ}}$ becomes active the Program Memory Page Select hardware produces page bits which correspond to the jump location. The full 12-bit address of the current instruction location is saved in the Return Address register. Resumption of normal program execution after completion of the interrupt service routine is controlled in software via the I/O return field of the auxiliary instruction bits. When the I/O return signal $\overline{\text{IOR}}$ becomes active another jump instruction is placed on the

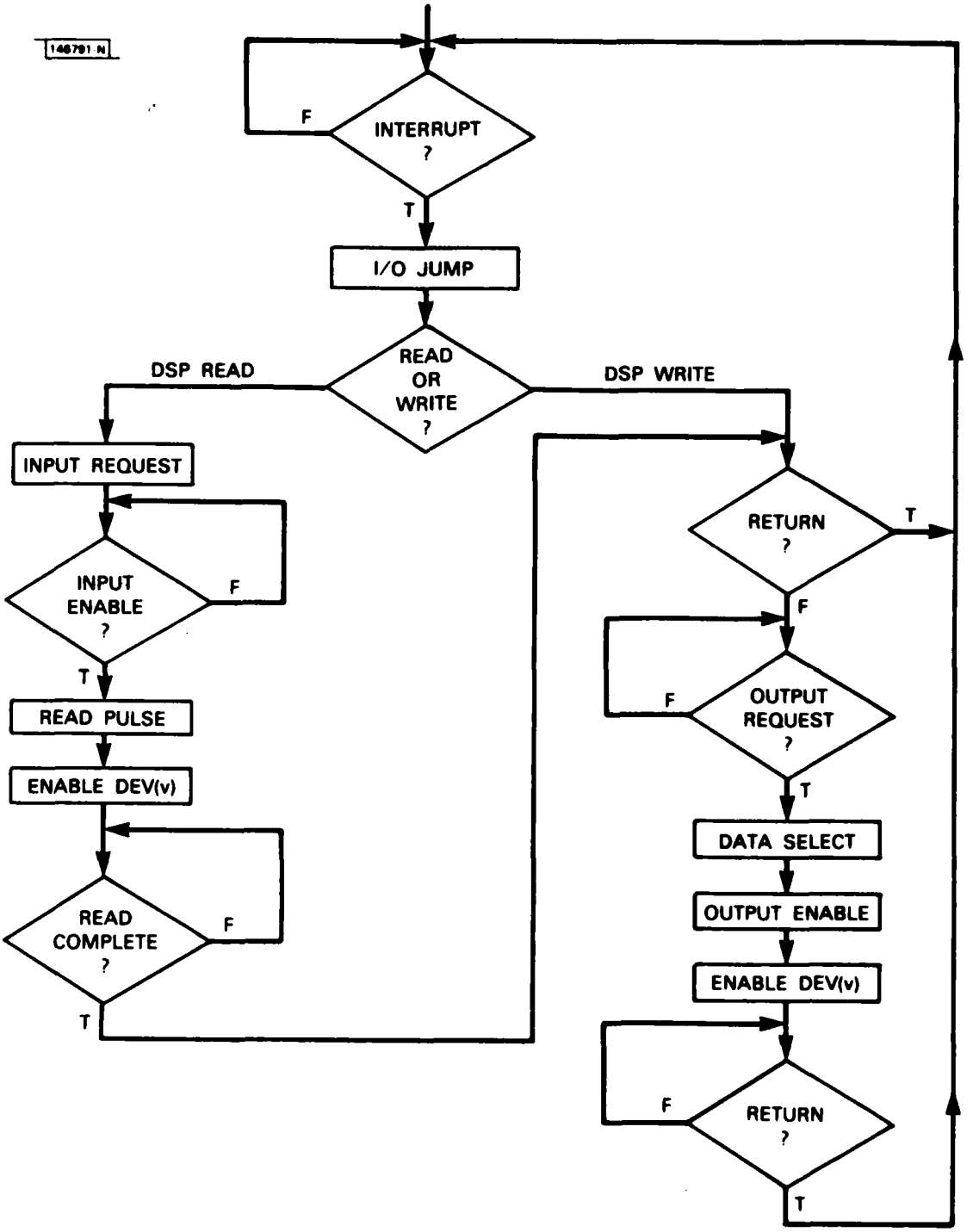



Fig. 4. Peripheral interface operation.

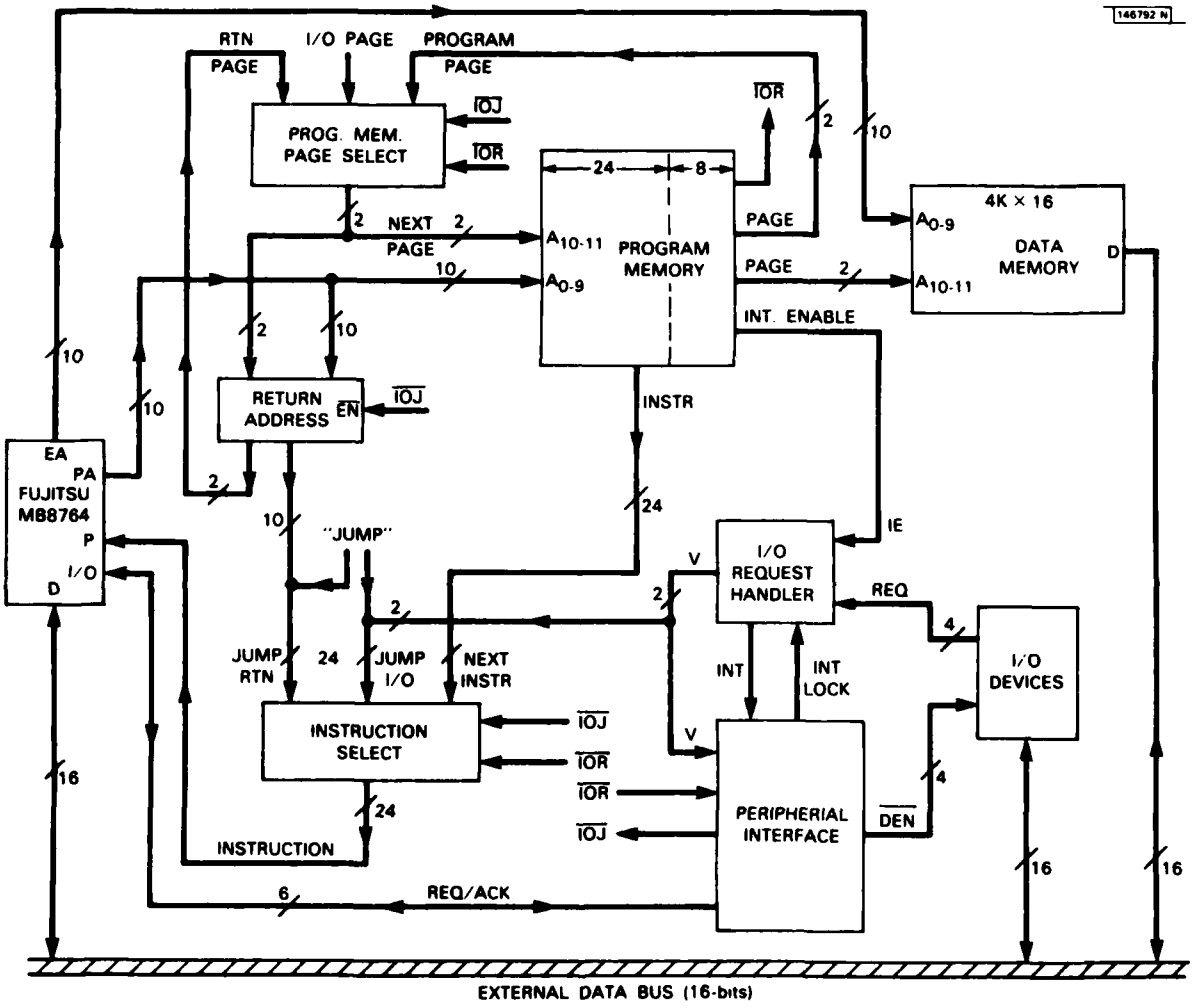



Fig. 5. Functional realization of interrupt controller.

instruction word input pins of the DSP with the contents of the jump address field coming from the low 10 bits of the Return Address register. The correct return page is supplied by the Page Select hardware.

The consequence of bringing the program and data address bits off chip is that the propagation delays inherent in the externally connected hardware will influence the maximum rate at which the processor can be run. For example, the MB8764 settling time for the data memory address is 55 ns and the set-up time for the data input is 22 ns. The corresponding times for the program memory path are 70 ns and 5 ns, respectively. Memory with extremely fast access times must be employed to achieve the 125 ns design goal of this project. Additional logic elements placed in these critical paths will inevitably require slowing down the system clock. It is for this reason that the processor was designed to implement the I/O jump through the instruction input pins rather than the program address output lines, thereby eliminating the need for tri-state buffers on the PA output pins.

A hardware realization of the processor is presented in Fig. 6. It is apparent that during a normal instruction sequence neither the program memory nor the data memory paths introduce any delays other than those associated with the memory access times. The Return Address Register (RAR) and Return Page Register (RPR) are enabled during an I/O jump in order to save the current program address. On the same cycle the appropriate control signals enable a vectored jump instruction onto the instruction lines (P) of the DSP. Two pipeline registers, DPR0 and DPR1, are provided for the data page bits so as to make the latency of this path commensurate

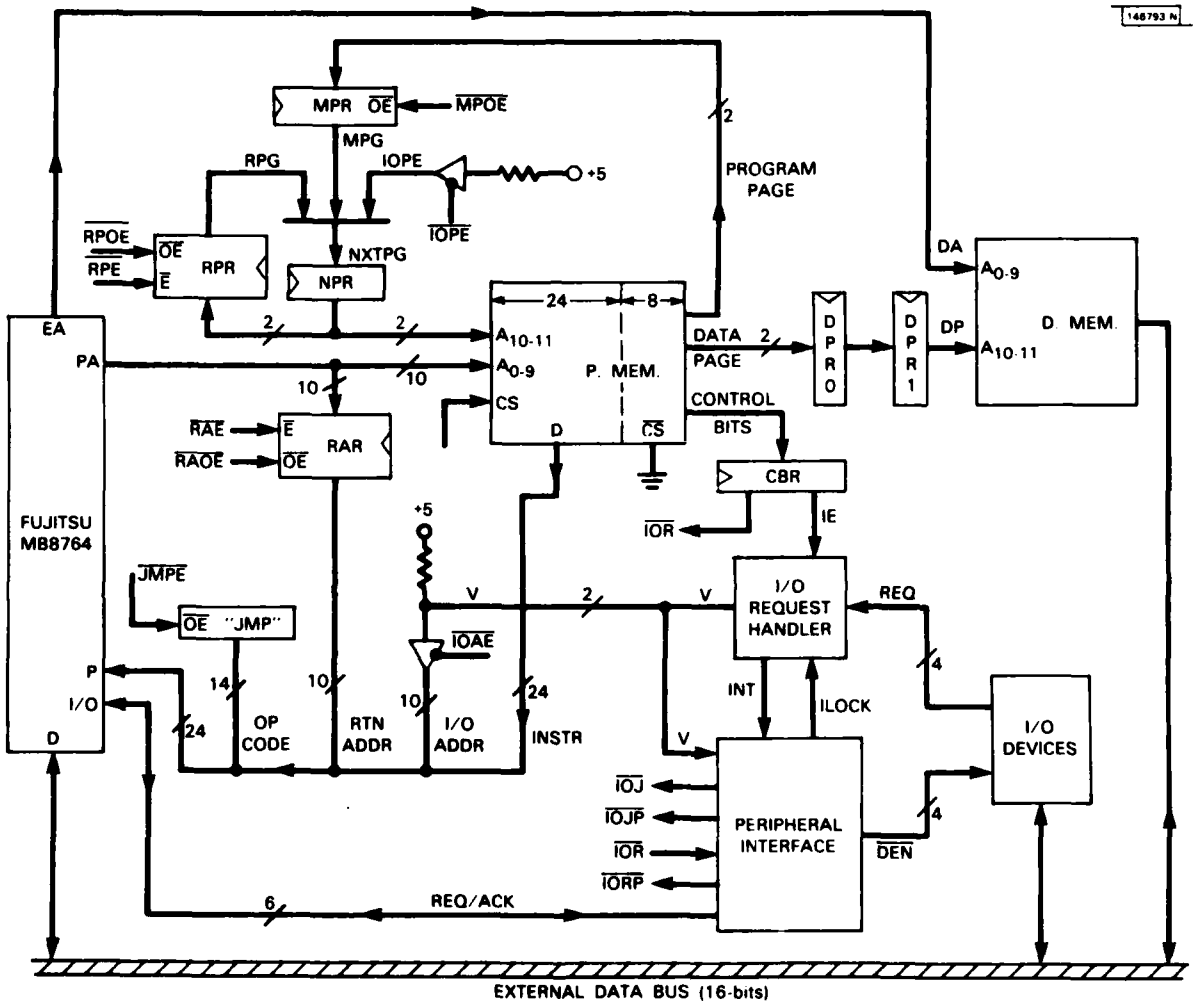



Fig. 6. Component level implementation of MB8764-based speech processor.

with the internal pipelining of the DSP. The program Memory Page Register (MPR) and Control Bits Register (CBR) allow the instruction path to be reclocked in order to meet the proposed 125 ns cycle time. The Next Page Register (NPR) is set up with the appropriate page bits, either the program page from the memory, the special I/O page containing the interrupt service routine jumps, or the saved return address page.

The Peripheral Interface is designed to supply the appropriate request and acknowledge signals between the DSP and the peripheral devices. Although commercially available PLAs and logic sequencers are well-suited to this task, they consume a considerable amount of power (about 800 mW maximum). Furthermore, at least two packages would be required to accommodate all the necessary input and and output signals. An alternative to this approach would be to design a custom chip using available VLSI tools. For example, the Lincoln Boolean Synthesizer (LBS) [9] has the capability of producing custom integrated circuits using low power, 3 micron CMOS technology. Rather than designing the chip directly, the feasibility of the custom approach was assessed using the MacPitts Silicon Compiler as a simulation vehicle [10]. The results indicated that a custom chip of dimensions 3.95 mm x 3.53 mm could be produced using 1104 transistors and dissipating 240 mW. Because MacPitts uses NMOS technology, the resulting chip would be too slow to operate at the 8 MHz frequency required by the processor. Using LBS, however, the chip could be realized with about 2200 transistors and would dissipate under 100 mW. Chips designed using LBS can be manufactured through the DARPA MOSIS silicon foundry service [11] in 28-pin double width packages.

In order to obtain a rough estimate of the size and power requirements of the processor, a component level design was developed and is summarized in Table V. Not all of the details have been thoroughly worked out, particularly with respect to the audio section and the clock logic. The power dissipation estimates for the registers and buffers were derived from specifications for National Semiconductor high speed CMOS components [12]. The I/O Request Handler implementation is similar to the one used in the Lincoln ALPCM processor [13] which required two PLAs and a register package. Since the finite state machine implemented in the I/O Request Handler is relatively simple, it is realistic to assume that its function could be combined with the Peripheral Interface into a single 28-pin LBS CMOS chip. The Peripheral Interface would probably require three packages if it were implemented using commercially available components. The total power requirements of the proposed processor are less than 10W (maximum) and less than 8W if the I/O Request Handler and the Peripheral Interface are realized in a single custom IC. The layout presented in Fig. 7 demonstrates the feasibility of implementing the hardware design on a single 7" x 7" wire wrap board.

## VI. SUMMARY

A study of the feasibility of realizing a compact, multiple function speech processor using digital signal processing integrated circuits is presented in this report. Designs based on the Texas Instruments TMS32010 and the Fujitsu MB8764 appeared to have the greatest likelihood of achieving this goal. An examination of the chips indicated that either one would require a significant amount of external support hardware in order to

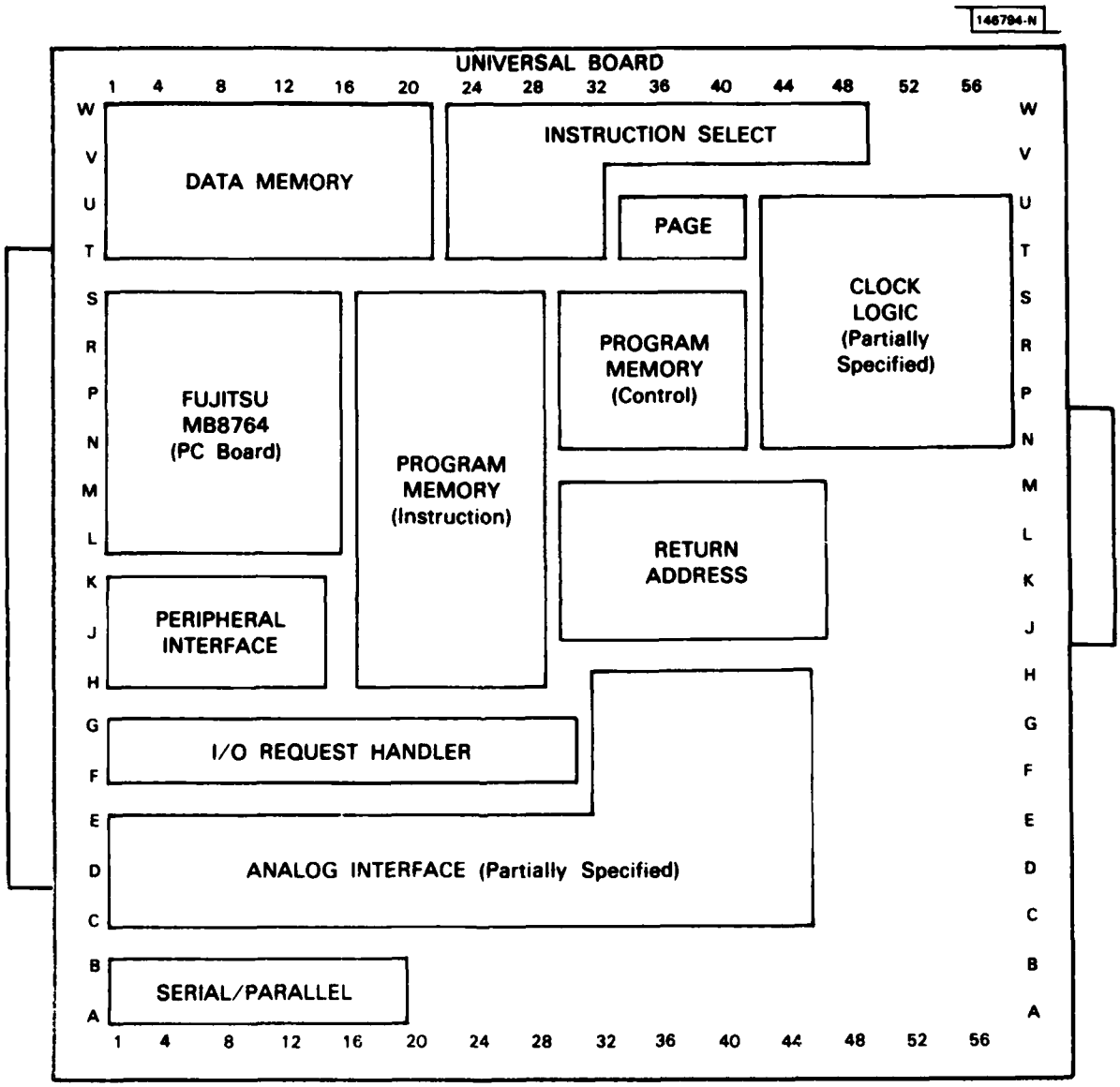



Fig. 7. Proposed processor layout.

TABLE V

SPEECH PROCESSOR COMPONENTS

| Function | Components | Packages | Power (mW) |
|---|---|---|---|
| Processor | Fujitsu MB8764 | 1 | 300 |
| Data Memory | Static RAM | 4 | 2000 |
| Program Memory -Instruction | PROM | 3 | 2775 |
| Program Memory -Control | Registered PROM | 2 | 1650 |
| Page Registers | Registers | 1 | 28 |
| Return Address Register | Registers | 4 | 68 |
| Instruction/Address Select | 3-State Buffers | 4 | 40 |
| I/O Request Handler | Registers, PALs | 3 | 1800 |
| Peripheral Interface | VLSI via LBS | 1 | <100 |
| Serial Interface | P/S, S/P Converters | 2 | 68 |
| Analog Interface | Codec, PSC and SPC, analog | ? | ~500 |
| Clocks | Crystal, logic | ? | ? |

be useful for sophisticated speech processing functions. Enhanced memory access capability is required by the TMS32010 to take full advantage of its fast multiply-accumulator. One realization of this goal is the APC processor designed at Lincoln Laboratory which contains automatic address generation logic designed specifically for sequential access of large blocks of external data. The study showed that the features incorporated in this design are well-matched to the computational and algorithmic requirements of the LPC-10 vocoder and the 9600 bps modem. However, real time processing of both the APC and modem functions appears to be met by relatively close margins.

The sophisticated architecture and faster cycle time of the Fujitsu MB8764 make this chip particularly well-suited to the speech processing applications of interest. Unfortunately, the lack of a hardware interrupt provision and the limited memory address space imply that special purpose hardware must be added to make the device useful for the current application. These drawbacks were overcome by implementing external interrupt hardware and by expanding both program and data memory using a paging scheme. An extended microinstruction word was used to provide the paging bits and selected control signals. To determine the feasibility of the architecture, a component level design was produced. Where possible, use was made of low power CMOS components to reduce power dissipation. The feasibility of implementing the hardware functions, particularly the finite state machines, with custom VLSI developed via in-house integrated circuit design tools was also explored and found to be advantageous. An MB8764-based processor appears to be well-suited to a compact, relatively low power implementation of a multiple function speech processor.

## ACKNOWLEDGEMENT

Many thanks to Ed Hofstetter and Joe Tierney who, as usual, provided their invaluable assistance during the course of this project.

## REFERENCES


[1] J.A. Feldman, E.M. Hofstetter, and M.L. Malpass, "A Compact, Flexible LPC Vocoder Based on a Commercial Signal Processing Microcomputer," IEEE J. Solid-State Circuits, SC-18, 4 (1983).

[2] A.W. Holck and W.A. Anderson, "A Single-Processor LPC Vocoder," IEEE Int. Conf. ASSP, March 1984, pp. 44.13.1-44.13.4.

[3] B. Bryden and H. Hassanein, "Implementation of a Full Duplex 2.4 kbps LPC Vocoder on a Single TMS-320 Microprocessor Chip," IEEE Int. Conf. ASSP, March 1984, pp. 44.12.1-44.12.4.

[4] M. Randolph, "The Design of an Adaptive Predictive Coder Using a Single-Chip Digital Signal Processor," Technical Report 679, Lincoln Laboratory, M.I.T., (to be published).

[5] E.M. Hofstetter et al., "Vocoder Implementations on the Lincoln Digital Voice Terminal," presented at the 1975 EASCON Conf., Washington, D.C., September 29-October 1, 1975.

[6] Fujitsu Limited, Fujitsu MB8764 General Purpose 16 Bit Digital Signal Processor (1983).

[7] R. Cohn, "Documentation Package for the NSA 9600 bps Software Wireline Modem," private communications (1981).

[8] R. Cohn, private communications.

[9] J.R. Southard, A. Domic, and K.W. Crouch, "LBS - Lincoln Boolean Synthesizer," Technical Report 622, Lincoln Laboratory, M.I.T. (1 September 1982), DTIC AD-A120999.

[10] J.M. Siskind, J.R. Southard, and K.W. Crouch, "Generating Custom High Performance VLSI Designs from Succinct Algorithmic Descriptions," Proc. Conf. on Advanced Research in VLSI (January 1982).

[11] G. Lewicki et al., "MOSIS: Present and Future," Proc. Conf. Advanced Research in VLSI (1984).

[12] National Semiconductor Corporation, MM54HC/74HC High Speed microCMOS Logic Family Databook (1983).

[13] E.M. Hofstetter, E. Singer, and J. Tierney, "A Programmable Voice Processor for Fighter Aircraft Applications," Technical Report 653, Lincoln Laboratory, M.I.T. (18 August 1983), DTIC AD-A133780/7.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>ESD-TR-84-283 | 2. GOVT ACCESSION NO.<br>A150 873 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>Comparative Architectures for a Multiple Function Speech Processor | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical Report<br>6. PERFORMING ORG. REPORT NUMBER<br>Technical Report 712 |
| 7. AUTHOR*(s)*<br>Elliot Singer | | 8. CONTRACT OR GRANT NUMBER*(s)*<br>F19628-85-C-0002 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Lincoln Laboratory, M.I.T.<br>P.O. Box 73<br>Lexington, MA 02173-0073 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>Program Element Nos. 63735F and 33401G |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Systems Command, USAF<br>Andrews AFB<br>Washington, DC 20334 | | 12. REPORT DATE<br>18 December 1984<br>13. NUMBER OF PAGES<br>44 |
| 14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*<br>Electronic Systems Division<br>Hanscom AFB, MA 01731 | | 15. SECURITY CLASS. *(of this report)*<br>Unclassified<br>15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

speech processor architectures
multiple rate processors
digital signal processing microcomputers
speech compression
speech processors
modems
compact implementations

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A study of the feasibility of realizing a compact, multiple function speech processor using digital signal processing integrated circuits is presented in this report. The processor is required to accommodate a 2400bps LPC10 vocoder, a 9600bps APC vocoder, and wireline modems at the corresponding data rates. A proposed architecture based on the Texas Instruments TMS32010 includes automatic address generation hardware to permit fast processing of large blocks of data stored in external memory. An architecture based on the Fujitsu MB8764 includes special purpose hardware which adds interrupt capability to the device. A detailed study of the Fujitsu-based design indicates that a compact realization of the multiple function processor can be realized using commercially available hardware and a custom IC fabricated using in-house design tools.

DD FORM 1473 1 Jan 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

END

FILMED

3-85

DTIC